

**REMARKS**

Applicants have now had an opportunity to carefully consider the Examiner's comments set forth in the Office Action of June 21, 2007. Claims 1-3 remain in this application. Claim 1 has been amended herein. No claims have been cancelled or withdrawn. Reconsideration of the Application is requested.

**The Office Action**

Claim 1 was rejected under 35 U.S.C. § 102(e) as being anticipated by Barry et al. (U.S. Pat. No. 7,046,391).

Claims 2 and 3 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Barry et al. in view of Bender et al. (U.S. Pat. No. 6,038,033).

**The Present Application**

By way of brief review, the present application addresses the continuing need for faster printing systems – particularly those where every page can be in different colors. More particularly, the subject invention comprises a method of improving efficiency of rasterizing image processing through a new implementation of parallelism. Prior implementations of parallelism in printing systems had been limited to job or page parallelism. However, job parallel processing results in poor single job performance, unpredictable job time and reduced throughput when there is only one long job in the queue. Page parallel processing is inefficient because per job overhead occurs on a per page basis. Either form of parallelism presents significant disadvantages in efficiency.

The method of the subject invention comprises a unique implementation of parallelism which Applicants have named "chunk" parallelism, an intermediate level of parallelism between job parallelism and page parallelism. A chunk is a collection of data describing an integer number of pages consisting of at least one page and not more than one job. This allows startup overhead in processing to occur on a per chunk basis, rather than on a per page basis, but prevents a single large job from reducing performance. The subject invention implements chunk parallelism in such a way as to

increase the efficiency of rasterizing image processing in a print system by tending to achieve an equal amount of processing work between parallel processing nodes in the print system, keeping multiple processors busy on a given job.

A preferred embodiment of the subject invention is a printing device comprising plural parallel rasterizing image processing nodes. Chunk size is determined by splitting factors for enhancing page processing efficiency. The splitting factors comprise either a threshold boundary of a number of bytes or a threshold boundary of a number of pages. In processing, the end of a chunk is the next page boundary after crossing a threshold boundary. The threshold boundaries for the number of bytes and the threshold boundary for the number of pages are determined for tending to achieve an equal amount of processing work between the plural processing nodes per boundary, and for keeping multiple processors busy on a given job, respectively. It is therefore, the flexibility of chunk size that gives chunk parallelism a significant advantage in efficiency over other forms of parallelism.

A key feature is virtual disk (VDISK) **38**, which is used for temporary storage, both of split chunks and print-ready pages. VDISK is similar to RAM disk with some specific features designed for performance in the contemplated chunk parallel system.

VDISK appears to both the sending and receiving processes like regular disk, with the ability to open files and directories, read and write files, etc. Unlike regular disk, VDISK provides the functionality that it may include a remote transfer (if the receiving process is on another node in a networked system), and because it knows whether the sending process has “closed” the file, it knows whether the receiving process should receive an end of file signal or be blocked on an attempt to read beyond the last data written. VDISK is implemented by a process providing a shared-memory interface for the receiving process, for local accesses. The VDISK implementation provides for more data being written to VDISK than fits into memory by paging out blocks of data in a most-recently-used order, because the least recently used (written) pages will be needed (read) soonest. Other than providing a transparent networked file system interface, blocking reads before end of file, and most-recently-used paging, VDISK operates much like a conventional RAM disk.

## The Cited Reference

In contrast, Barry et al. is not concerned with improved performance in parallelism over job parallelism or page parallelism. Instead, Barry et al. describes an implementation of page parallelism without questioning its efficiencies or inefficiencies. For instance, Barry et al. states:

"[The job] is then passed to a software RIP engine which is operable to essentially decode the print string that is received from the print spooler. This effectively divides each print job into pages. These pages are then stored in page buffers. Each page in the page buffer essentially constitutes a single print job, such that any print job received from the workstations 10 will then be parsed into a multiple print job file."

Moreover, Barry et al. never suggests that page parallelism is inefficient or undesirable for that very reason. Indeed, as Barry et al. addresses the problems of color balancing within jobs received from multiple sources and spread over multiple print engines, the page parallel setup may be desirable or even necessary. It is important, therefore, to note that Barry simply does not discuss alternative forms of parallelism, either as methods for increasing the efficiency of rasterizing image processing or otherwise.

Barry, when using the term 'virtual', refers to a concept in which multiple physical printers behave as if they were a single printer - that is a virtual printer is made up of multiple physical printers; and also a 'virtual stack', which is a concept that emulates a stack of paper. Barry rasterizes the entire job and then splits up the rasterized portions and sends them to various print engines, so that when the operator subsequently piles the output of one engine atop the output of another engine, the resulting stack looks like what would have happened had the request been sent to a single engine.

Nowhere does Barry introduce the concept of a virtual disk. A virtual disk is similar to a RAM disk, in that it behaves like very fast disk - the application programmer uses the same interface as if it were a disk - however in this context it is stored in RAM, unless it overflows, in which case it goes to physical disk, writing the newest portion to disk first. This is totally unanticipated by Barry. VDisk is an important innovation as it

enables a parallel RIP to run much faster than the prior art methods, such as Adobe Extreme, which required the physical disk.

### **The Claims Distinguish Over the Teachings of the References**

The Examiner will appreciate that claim 1 has been substantially amended to more particularly emphasize the novel aspects of the present embodiments over the teachings of Barry. In particular, the VDISK featured has been specified to be a composite of both a RAM and a physical disk in that the RAM receives data, to achieve data expediency advantages until such time it is essentially full, then newer data is first written to the physical disk, until the RAM becomes available again. In the context of a transfer system where the data either is chunk data or print-ready pages, the data needs to be transferred out to any of the selective processing nodes, and transferring from a RAM, as opposed to the physical disk, enables the parallel RIP processing to run much faster. Claim 1 now includes the steps of detecting the data overflow in the RAM and then storing new data in the physical disk so that this selectively storing of the job chunks in the virtual disk transfer system can achieve these desired expedited handling advantages.

Barry's teaching of a virtual stack is not the same as the subject embodiments innovation of a VDISK comprised of a RAM and a physical disk operating in cooperation for a chunk storage to expedite chunk data transferred to processing nodes. Nowhere in Barry is there a suggestion that the virtual stack is monitored as to any data storage overflow. The concept of a virtual stack is quite different in purpose and structure so that an analogy of a virtual stack to the method of the subject embodiments is difficult to precisely traverse, other than to say that the two concepts are different.

### **CONCLUSION**

Claims 1-3 remain in the application. Claim 1 has been amended. For at least the reasons detailed above, Applicants submit that all claims remaining in the application (Claims 1-3) are now in condition for allowance. Accordingly, Applicants request an early indication thereof. The foregoing comments do not require unnecessary additional search or examination.

No additional fee is believed to be required for this Amendment A. However, the undersigned attorney of record hereby authorizes the charging of any necessary fees, other than the issue fee, to Xerox Deposit Account No. 24-0037.

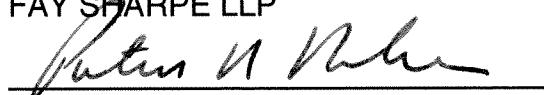
In the event the Examiner considers personal contact advantageous to the disposition of this case, he/she is hereby authorized to call Patrick R. Roche, at Telephone Number (216) 861-5582.

Respectfully submitted,

FAY SHARPE LLP

November 21, 2007

Date

  
\_\_\_\_\_  
Patrick R. Roche, Reg. No. 29,580  
1100 Superior Avenue, Seventh Floor  
Cleveland, OH 44114-2579  
216-861-5582